In the claims:

For the Examiner's convenience, all pending claims are presented below with changes shown.

1. (Currently Amended) A method comprising:

inserting a single instruction at ~~a start~~the block header of a block of code ~~to run~~ configured for execution on a first processor architecture, wherein the single instruction is to determine if processor resources needed to execute the block of code are available;

emulating the block of code on a computer system configured for execution on a second processor architecture;

using the single instruction to monitor the resources of the computer system ~~second processor~~ used during emulation to determine whether resource requirements of the first processor architecture have been exceeded; ~~and~~

if the resource requirements of the first processor architecture have been exceeded, modifying allocation of the resources of the computer system ~~second processor~~ according to the resource requirements of the first processor architecture; and

if the resource requirements of the first processor architecture have been not exceeded, continuing emulation of the block of code.

2. (Currently Amended) The method of claim 1, further comprising:

determining a set of available resources that will be available after ~~said~~ the block of code has executed.

3. (Canceled)

4. (Currently Amended) The of claim 1 wherein the availability of the ~~processor~~ resources of the computer system is determined at compile time.

5. (Currently Amended) The method of claim 1 wherein the availability of the ~~processor~~ resources of the computer system is determined dynamically.

6. (Currently Amended) The method of claim 1 further comprising:

signaling an error message if the resources of the computer system ~~of the processor~~ needed ~~for~~ to execute the block of code are not available; and

in response to the error message, branching to a fault handler routine.

7. (Currently Amended) The method of claim 6 wherein signaling of ~~said~~ the fault handler routine simulates a processor exception.

8. (Currently Amended) The method of claim 1 wherein the resources of the computer system are represented by a bit vector.

9. (Currently Amended) The method of claim 8 wherein ~~said~~ the bit vector is generated dynamically.

10. (Currently Amended) A computer-readable medium having stored thereon a set of instructions to monitor processor resources, said set of instruction, which when executed by a processor, cause said processor to perform a method comprising:

inserting a single instruction at ~~a start~~ the block header of a block of code ~~to run~~ configured for execution on a first processor architecture, wherein the single instruction is to determine if processor resources needed to execute the block of code are available;

emulating the block of code on a computer system configured for execution on a second processor architecture;

using the single instruction to monitor the resources of the computer system ~~second processor~~ used during emulation to determine whether resource requirements of the first processor architecture have been exceeded; ~~and~~

if the resource requirements of the first processor architecture have been exceeded, modifying allocation of the resources of the computer system ~~second processor~~ according to the resource requirements of the first processor architecture; and

if the resource requirements of the first processor architecture have been not exceeded, continuing emulation of the block of code.

11. (Currently Amended) The computer-readable medium of claim 10, wherein said set of instructions further includes additional instructions, which when executed by said processor, cause said processor to further perform said method comprising:

determining a set of available resources that will be available after ~~said~~the block of code has executed.

12. (Canceled)

13. (Currently Amended) The computer-readable medium of claim 10 wherein the availability of the ~~processor~~ resources of the computer system is determined at compile time.

14. (Currently Amended) The computer-readable medium of claim 10 wherein the availability of the ~~processor~~ resources of the computer system is determined dynamically.

15. (Currently Amended) The computer-readable medium of claim 10 wherein additional instructions, which when executed by the processor, cause the processor to perform the method further comprising:

signaling an error message if the resources of the computer system ~~the processor~~ needed ~~for~~ to execute the block of code are not available; and

in response to the error message, branching to a fault handler routine.

16. (Currently Amended) The computer-readable medium of claim 15 wherein signaling of ~~said~~ the fault handler routine simulates a processor exception.

17. (Currently Amended) The computer-readable medium of claim 10 wherein the resources of the computer system are represented by a bit vector.

18. (Currently Amended) The computer-readable medium of claim 17 wherein ~~said~~ the bit vector is generated dynamically.

19-25. (Canceled)